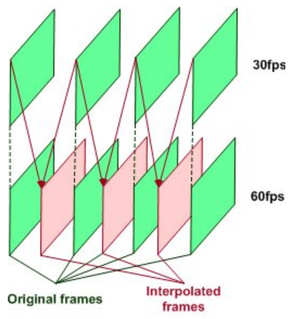


Frame Rate Upscaling

Gautham Kesineni, Ted Xiao, Raul Puri

Berkeley EECS
Electrical Engineering & Computer Sciences

Introduction

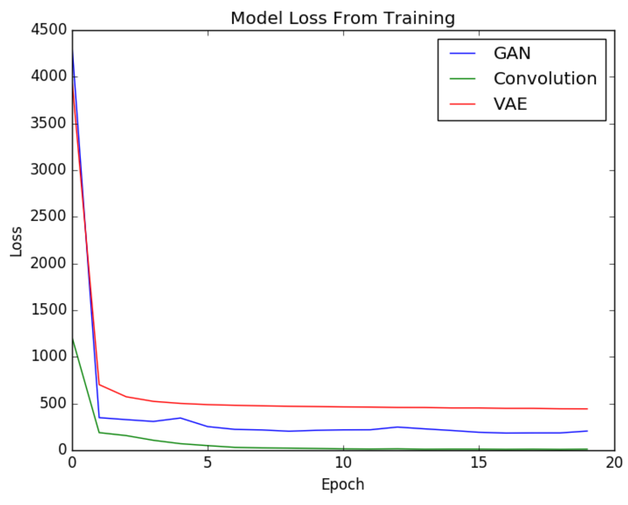
Frame interpolation is a computer vision task that is largely performed on real world video data. Interpolating intermediate frames between given video frames would produce an increase in frames per second (FPS). There are many potential applications of frame rate upscaling, but we see the biggest impact in video streaming: this technique could allow for the streaming of lower frame rate video that could then be interpolated to generate more frames and produce smoother high-framerate video.

State of the Art

Traditional frame interpolation usually separates the image into components to measure the motion vectors of the image. These vectors are used to find the location of different blocks at different points in time. This generally works well but is not ideal. The algorithm knows nothing about the way objects move in the real world and assumes all motion is linear. When frame rates are very low, the artifacts of this method start appearing. For this reason, we believe there is great promise in deep learning (DL) methods for this problem. Currently, state of the art DL models leverage variational autoencoders to achieve this.

Methods

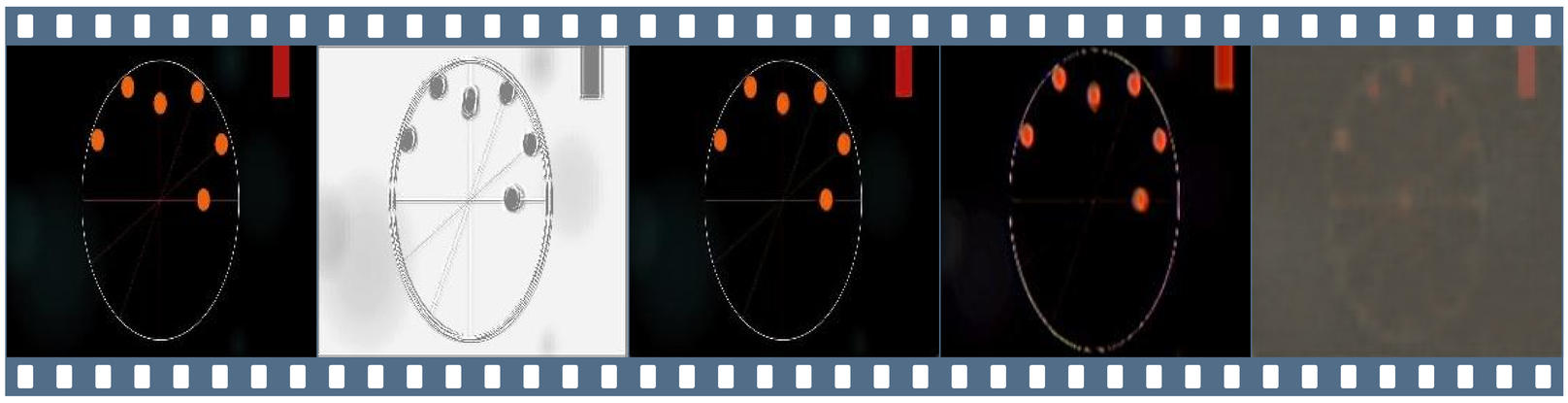
We tried several methods in our approach: Simple Convolution (Baseline), Deep Convolution, Variational AutoEncoders, and GANs. More details on these along with the results can be found in this poster. Below is the loss over time while training each of these models.

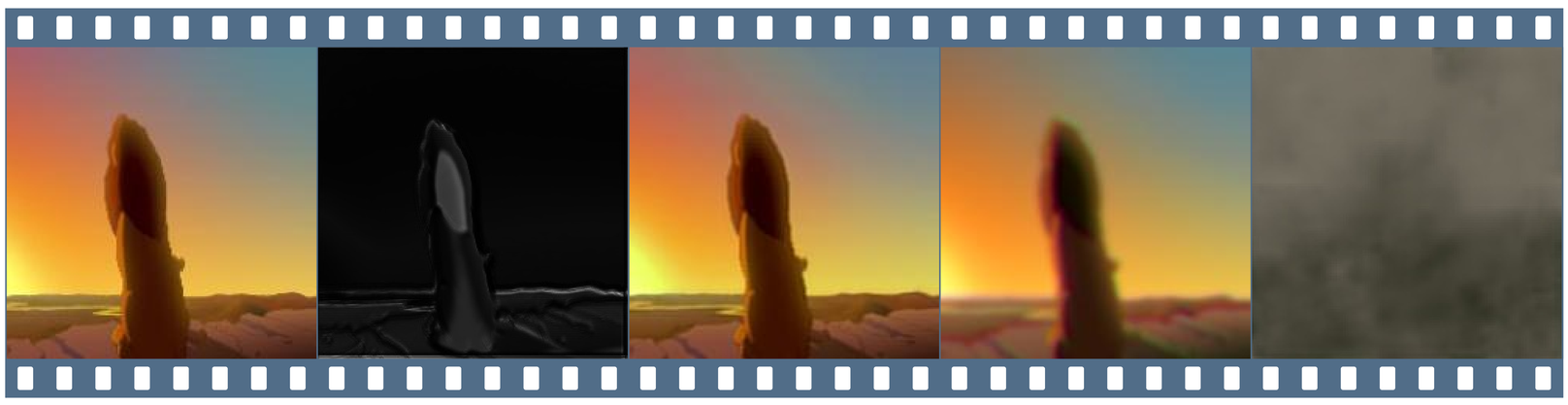


Conclusion

Throughout the course of this project we experimented with numerous techniques. Given more time we'd like to experiment more with recurrence and longer windows of prediction, experiment with residual connections in our networks and possibly a U-Net style VAE variant in order to mitigate information loss. We learned a lot from this class and our project. We hope to continue to refine our results to see if we can publish.

Results





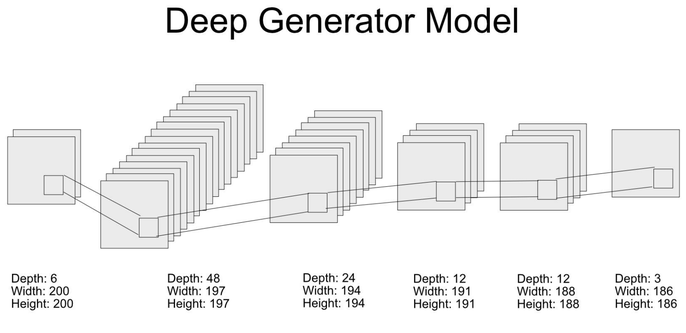
The deep convolution network performed remarkably well for its simplicity. We found that this network performed especially well when the camera was moving but less so when objects within the scene moved. The GAN network generalized better. The baseline model essentially performed linear gray scale interpolation. The VAE tried to memorize the image during the compression, and merely scales up subsections of the image.

Baseline Model

Our baseline model was a simple 3-layer convolution. The biggest change we made here was to use $\tanh$ instead of ReLu functions and to normalize the inputs.

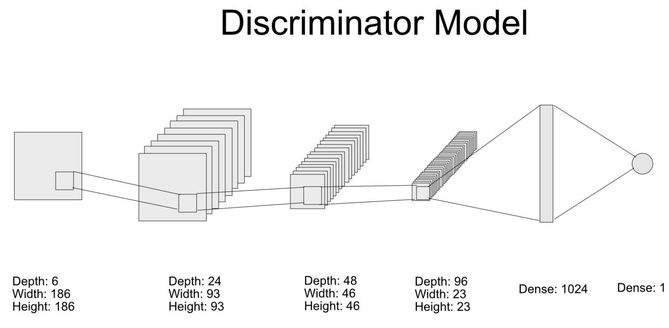
Deep Convolution Model

This was an extension of our first model. We increased the number of layers to 6. The biggest change is that we swapped the $\tanh$ activations with ReLu and used a linear activation for our last layer. We found that $\tanh$ would favor the values at 1 or -1 while a linear activation would treat all equally, as pixel values should be.



GAN Model

The Generative Adversarial Network (GAN) model is an extension of our Deep Convolution Model. Unlike typical GANs where the input is random bits, we supply the input as the before and after frames and attach a normal discriminator component. We applied both MSE and GAN updates to the generator component to converge faster.



VAE model

We tried using Variational AutoEncoders (VAEs) in our experiments. Due to the compressive nature of VAEs our results generally ended up being rather blurry as shown.